



1/11

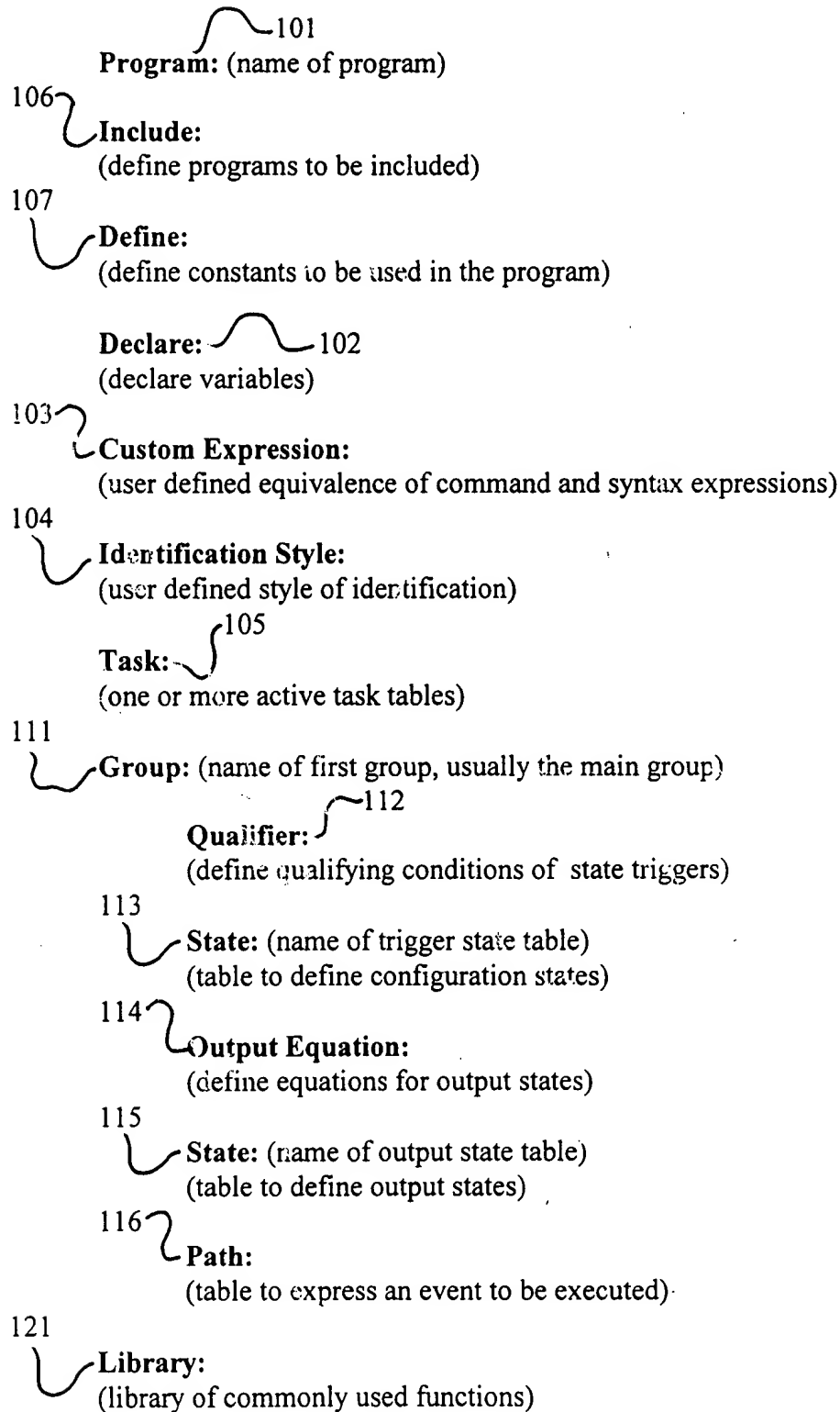


Figure 1



2/11

201 Custom Expression: MySign 210
202 (& &) = (AND) 203 // logical AND function 204
205 (||) = (BIT OR) // bit to bit OR function
206 (;) = (//) // comment notification
207 (++) = (INC) // increment
208 (! =) = (UNEQ) // not equal
--- --- ---
{ (%) = (REM) // remainder
209

200

Figure 2A

216
Path_DisplayA_1: RegisterA (C : & &) 00001111; Display: RegisterA
Path_DisplayA_2: A: (RegisterA & 00001111); Display: RegisterA
217

Figure 2B



231 Identification Style: MyStyle 232
// This portion defines the style and appearance of either labels or keywords. 230
233
234 Case = Title 235 // select title case for keywords
Style = bold, italic, underline // define identified wordings to be of
239 240 // bold, italic or underline type

Figure 3

261	262	263	264	260
Task: ActiveTask	Main	Program2	Program3	--- Program n
Task Status 1	265 Start	266 x	x	x
Task Status 2	267 Run	Start	x	x
---	--	--	--	--
Task Status m	268 Run	Run	Run	Run
Task Priority 1	273 1	3	2	6 271
Task Priority 2	274 1	2	6	3 272

Figure 4



4/11

Task: Input	Keyboard	Mouse	GamePort
All:	Run	Run	Run
Normal:	Run	Run	x
Keyboard:	Run	x	x
Mouse:	x	Run	x
Game:	x	x	Start
Task: Ports	COM1	COM2	ParallelPort
All:	Run	Run	Run
SrMouse:	Start	x	x
ExModem:	x	Continue	x
Printer:	x	x	Continue
Task: Device	CDRom	HardDriveC	Floppy
ReadCD:	Start	Run	x
HDFullSpeed:	x	Run	x
RWFloppy:	x	Run	Run
ALL:	Run	Run	Run

//Possible task states: Start, Continue, Run, Pause, x

Figure 5A



5/11

301 Task: Input	302 Keyboard	303 Mouse	304 GamePort
305 All:	Run	Run	Run 310
306 Normal:	Run	Run	x
307 Keyboard:	Run	x	x
308 Mouse:	x	Run	x 311
Game: 309	x	x	Start
321 Task: Ports	322 COM1	323 COM2	324 ParallelPort
325 All:	Run	Run	Run
326 SrMouse:	Start	x	x
327 ExModem:	x	Continue	x 329
Printer: 328	x	x	Continue
341 Task: Device	342 CD Rom	343 HardDriveC	Floppy
345 ReadCD:	Start	Run	x 344
346 HDFullSpeed:	x	Run	x
347 RWFloppy:	x	Run	Run
ALL: 348	Run	Run	Run
349 //Possible task states: Start, Continue, Run, Pause, x			

Figure 5B



6/11

```

1  Group: Main                //WebSale program main group

2  Qualifier:
3  Catalog = Icon(1)
4  Purchase = Icon(2)
5  Service = Icon(3)
6  Home = Icon(4)
7  Quit = Icon(5)

8  State: FirstPage   Catalog   Purchase   Service   Home   Quit
9  Ready:             P_catalog P_purchase P_service Start   Bye
10 Hold1:             x         x         x         Start   Bye

11 State: Response   Group:Info   Group:Order   Group:Service   Group:Register P3.1
12 WindowCatalog:   Run         x         x         x         x
13 WindowPurchase:  x         Run         x         Run         x
14 WindowService:   x         x         Run         Run         x
15 Hold2:           x         x         x         x         x
16 Hold3:           x         Run         x         Run         x
17 Beep:            Continue   Continue   Continue   Continue   P+

18 Path:
19 Start:           CheckSystem; DisplayFirstPage; Beep; Hold 2; Ready; END
20 P_catalog:       WindowCatalog; BuySolicit; END
21 P_purchase:      WindowPurchase; Hold1; Hold3; GrayButton; END
22 P_service:       WindowService; BuySolicit; END
23 Bye:            Terminate; END

24 EOG              //Keyword to indicate end of Group

```

//Note: Line numbers are for description reference and are not required in actual
//programming. States and path equations are not required to be in sequence.

Figure 6A



7/11

```

1  Group: Main                //WebSale program main group

2  Qualifier:
3  Catalog = Icon(1)
4  Purchase = Icon(2)
5  Service = Icon(3)
6  Home = Icon(5)
7  Quit = Icon(4)

8  State: FirstPage   Catalog   Purchase   Service   Home   Quit
9  Ready:             P_catalog P_purchase P_service Start   Bye
10 Hold1:             x         x         x         Start   Bye

11 State: Response   Group:Info   Group:Order   Group:Service   Group:Register P3.1
12 WindowCatalog:   Run           x             x             x             x
13 WindowPurchase:  x             Run           x             Run           x
14 WindowService:   x             x             Run           Run           x
15 Hold2:           x             x             x             x             x
16 Hold3:           x             Run           x             Run           x
17 Beep:            Continue    Continue     Continue     Continue     P+

18 Path:
19 Start:            EJ_CheckSystem; EVB_DisplayFirstPage; Beep; Hold 2; Ready; END
20 P_catalog:        WindowCatalog; Lbry:BuySolicit; END
21 P_purchase:       WindowPurchase; Hold1; Hold3; Lbry:GrayButton; END
22 P_service:        WindowService; Lbry:BuySolicit; END
23 Bye:              EJ_Terminate; END

24 Library: Local
25 BuySolicit:       EC_CheckRecord; EVC_SolicitWindow; END
26 GrayButton:       EC_CheckX; EVB_X_Is_Gray; END

27 EOG              //Keyword to indicate end of Group

```

//Note: Line numbers are for description reference and are not required in actual
//programming. States and path equations are not required to be in sequence.

Figure 6B



8/11

Include:

ReadIR	//Read signal from remote control
ReadPanel	//Read signal from control panel
ReceiveFromLine	//Input data from cable line
SendToLine	//Transmit data to cable line
CompressBuffer	//Compress data before send to cable line
DecompressBuffer	//Decompress data file
CompareTime	//Compare register time with real time clock
GenIcon	//Generate Icons onto TV screen
GenWindow	//Create a new window
InputCursor	//Input and decode movement of cursor
OutputCursor	//Update cursor position on TV screen
DisplayData	//Display data to TV screen
DisplayPicture	//Display picture file to TV screen
.....	//.....
.....	//.....
ErrorMessage	//Decode error code and transmit error message
Help	//Display help screen

Figure 7



9/11

Include:

EA_ReadIR	//Read signal from remote control
EE_ReadPanel	//Read signal from control panel
EA_ReceiveFromLine	//Input data from cable line
EA_SendToLine	//Transmit data to cable line
EC_CompressBuffer	//Compress data before send to cable line
EC_DecompressBuffer	//Decompress data file
EA_CompareTime	//Compare register time with real time clock
EJ_GenIcon	//Generate Icons onto TV screen
EJ_GenWindow	//Create a new window
EVC_InputCursor	//Input and decode movement of cursor
EVB_OutputCursor	//Update cursor position on TV screen
EVB_DisplayData	//Display data to TV screen
EVB_DisplayPicture	//Display picture file to TV screen
.....	//.....
.....	//.....
EA_ErrorMessage	//Decode error code and transmit error message
EVB_Help	//Display help screen

Figure 8



10/11

Function

Instruction separator

Identify end of instruction

Continuation of Instruction

Identify line continuation

Unary Operators

Direction

Address

Negative

Logical negation

1's complement

Increment

Decrement

Size of

Shift Operators

Left shift

Right shift

Relational Operators

Less than

Greater than

Less or equal

Greater or equal

Logical operators

Bitwise AND

Logical AND

Exclusive OR

Bitwise OR

Logical OR

Conditional branch

Comments

Table Format Instruction

;

& (at the beginning of a line)

VALUE (address)

ADDRESS (lvalue)

-

NOT (expression)

COMPLEMENT(expression)

INCREMENT

DECREMENT

SIZEOF(expression)

BIT SHIFT LEFT (operand, n)

BIT SHIFT RIGHT (operand, n)

<

>

<= or =<

>= or =>

BIT AND

AND

XOR

BIT OR

OR

Expression?: [True-statement] / [False-statement]

// (to the end of the line)

Figure 9



11/11

Input State:
Trigger Pin: TG1 TG2 TG3 TG4 TG5 TG6
State0: R:Path1 R:Path2 R:Path3 R:Path4 X X
State1: F:Path11 R:Path2 R:Path3 R:Path4 X X
State2: R:Path1 F:Path11 R:Path3 R:Path4 X X
State3: R:Path1 R:Path2 F:Path11 R:Path4 X X
State4: R:Path1 R:Path2 R:Path3 F:Path11 X X
;
Path:
Path 1: State1 Sound1 Path1
Path 2: State2 Sound2 Path2
Path 3: State3 Sound3 Path3
Path 4: State4 Sound4 Path4
Path11: State0 End

Figure 10 Prior Art

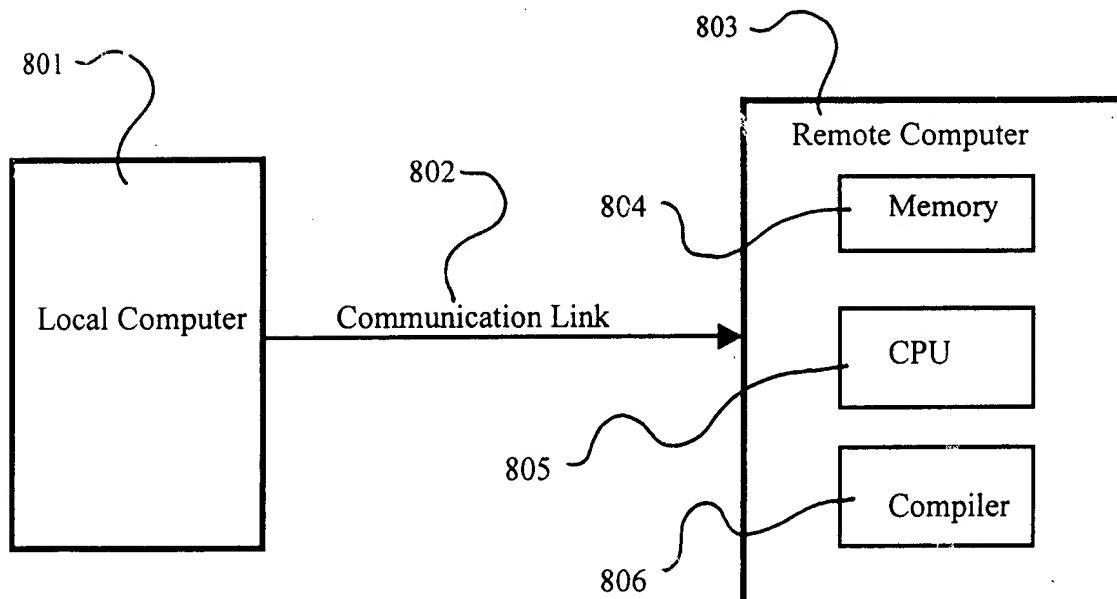


Figure 11